



WORKSHOP **TRAINER** MANUAL

Microsoft

PRESENTED BY
 **ideaslab**
EXPLORING NEW POSSIBILITIES

Módulo 1 Introducción al Diseño de Juegos y Kodu Game Lab | 3

Actividad 1.1 Preséntese | 4

Actividad 1.2 Introducción a Kodu Game Lab |4

Actividad 1.3 ¿Qué es un juego? | 4

Actividad 1.4 Jugos Kodu | 6

Módulo 2 Bases de Kodu |8

Actividad 2.1 Introducción al Lenguaje de Programación Kodu | 8

Actividad 2.2 Resolviendo Tutoriales 2 y 3 |10

Actividad 2.3 Conociendo los robots y los Objetos |13

Actividad 2.4 Un juego de dos robots |16

Módulo 3 Proceso de Diseño informal |17

Actividad 3.1 Generando Ideas | 17

Actividad 3.2 Ideas de Juego |17

Actividad 3.3 Ejemplos Escolares |18

Diseñando plantillas | 19

Módulo 4 Creando Mundos

Actividad 4.1 Diseñando Espacios de Juego |20

Actividad 4.2 Recetas y Patrones de Diseños de Mundos | 20

Actividad 4.3 Patrones de Espacios de Juego | 21

Módulo 5 Patrones de Diseño de Juegos |28

Actividad 5.1 Patrones de Diseño de Progreso de Juegos | 28

Actividad 5.2 Patrones de Diseño de Jugabilidad de Juegos| 38

Actividad 5.3 Patrones de Diseño de Información de Juego | 44

Module 6 Next Steps |45

Actividad 6.1 Diseño de Proceso Iterativo | 45

Actividad 6.2 Herramientas de Depuración, Solución de Problemas y Desarrollo | 46

Módulo 7 Próximos Pasos |48

Introducción

Kodu Game Lab es un lenguaje de programación visual que permite a cualquier persona, incluso sin experiencia ni conocimiento de programación, crear juegos 3D.

Kodu utiliza íconos intuitivos para controlar los personajes y objetos con comportamientos de personajes expresados en términos físicos.

El Curso de Creación de Juegos con Kodu Game Lab es diseñado básicamente para maestros que buscan introducir el diseño de juegos a su programa de clases. Este curso intensivo de dos días hará introducción a Kodu Game Lab y demostrará cómo cualquier persona puede crear entretenidos y elaborados juegos.

Este curso cubrirá las habilidades necesarias para diseñar y crear mundos y juegos con Kodu Game Lab y también investigará el proceso de diseño y cómo Kodu Game Lab puede ser utilizado en el expediente escolar.

A través del diseño y la creación de sus propios juegos, los participantes aprenderán cuan poderoso e interesante pueden ser creados los juegos con técnicas y bloques de construcción sencillos.

Este curso:

- Permitirá a los participantes programar personajes y objetos
- Permitirá a los participantes crear mundos en 3D
- Hará introducción a los elementos clave del diseño de juegos • proveerá una buena variedad de códigos de ejemplo
- Proveerá una buena variedad de juegos de ejemplo
- Proveerá ejemplo de cómo Kodu Game Lab ha sido utilizado en escuelas
- Dará muchas manos a la obra para que los participantes tengan confianza en utilizar Kodu Game Lab

Nota Importante

Kodu Game Lab aún está bajo desarrollo. Se ha hecho todo esfuerzo para asegurarse de que los ejemplos contenidos en este manual funcionen con la versión 1.48 de Kodu Game Lab. Al utilizar otras versiones, por favor acceda a <http://media.planetkodu.com/workshop/resources.html> Para actualizaciones y materiales de trabajo.

Este es el manual del entrenador. Las notas exclusivas para el entrenador aparecen en las cajas en todo el manual. Estas cajas no están incluidas en el manual del participante.



Módulo 1 Introducción al Diseño de Juegos y a Kodu Game Lab

Actividad 1.1 Preséntese

Use esta actividad Para que el grupo se pueda conocer. Permita 10 minutos para que el grupo cree sus personajes y después 10 minutos para que los compartan con los otros grupos

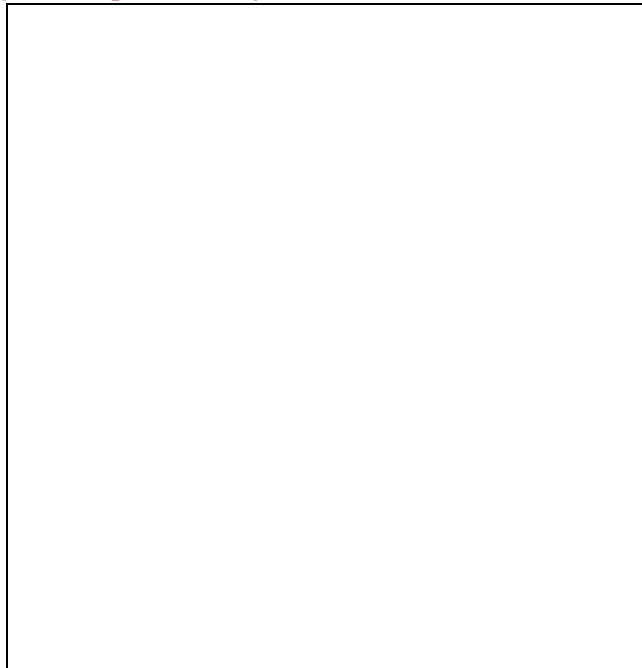
Si fueses un personaje de videojuegos: ¿Cómo serías?

Nombre: _____

Poderes: _____

3 cosas que harían especial a tu personaje:

Dibuja a tu personaje en el espacio de abajo:



Actividad 1.2 Introducción a Kodu Game Lab

Puntos claves para comunicar:

Kodu es para desarrollo rápido de juegos utilizando un interfaz y lenguaje.

Kodu utiliza un lenguaje de programación basado en íconos o metáforas físicas.

Kodu es para juegos pequeños

Durante esta actividad exploraremos Kodu Game Lab, su propósito y lo que otros han creado con él. Utilice el espacio de abajo para escribir cualquier idea importante sobre Kodu Game Lab.

Actividad 1.3 ¿Qué es un juego?

Pídale al grupo que piense en cinco juegos que le gustaría jugar. Motívelos a listar diferentes tipos de juegos, por ejemplo, un deporte, un juego de mesa, un juego de computadora. Una vez hayan escogido cinco juegos, pídale que piensen qué tienen en común, y qué conforma un juego. Utilice el diamante 9 para mostrar los factores clave sobre qué conforma un juego con los factores más importantes en la cima.

Los participantes deben compartir sus ideas al final de la actividad.

Puntos claves que recalcar:

Los juegos son divertidos.

Los juegos tienen reglas.

Los juegos tienen objetivos / Los juegos tienen ganadores y perdedores.

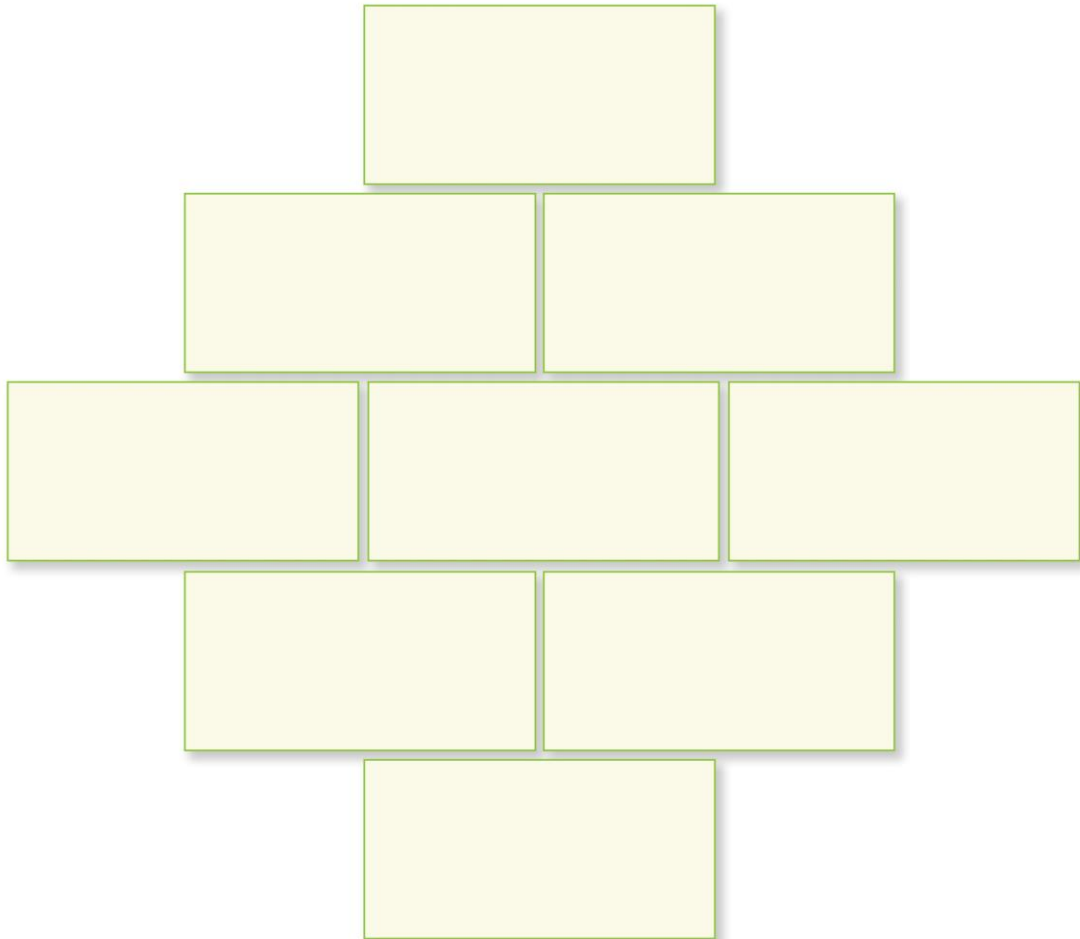
Los juegos requieren que los jugadores tomen decisiones.

Requieren habilidad. Los juegos son una experiencia.

En esta actividad exploraremos qué es lo que hace a un juego, un juego

Lista 5 juegos que te gustaría jugar:

Pensando sobre los juegos que haz listado, Identifica los factores clave que hacen a un juego un juego. Utiliza el organizador gráfico de 9 diamantes debajo para mostrar nueve factores de juegos con los factores más importantes en los diamantes de la cima.



“Si has escrito una subrutina de software que tome más de diez argumentos, observa de nuevo. Tal vez te faltaron algunos. ”

Alan Kay

“Un juego es una actividad de problema-solución, de la mano con una actitud juguetona.”

Jesse Schell

Actividad 1.4 Juegos Kodu

Esta actividad es diseñada para permitir a los participantes descubrir los tipos de juegos que pueden ser creados con Kodu Game Lab e identificar qué hace a estos juegos divertidos. Establezca 15 minutos para jugar., 5 minutos para escribir sus descubrimientos y 10 minutos para discusión. Use el Power Point de “Kodu Game”, para compartir juegos que han sido creados por la comunidad Kodu Xbox.

Como ejemplo, tenga una partida rápida de Bonk Out 18 y discuta sus objetivos, reglas y porqué es divertido

Kodu Game Lab tiene alrededor de cincuenta mundos y juegos incluidos por defecto. En esta actividad exploraremos algunos de estos mundos y juegos para aprender sobre lo que se puede hacer con Kodu Game Lab y qué hace a un juego de Kodu Game Lab.

Revisión de Juego (Ejemplo)

Nombre: Bonk-Out v18

Objetivo: Derriba los castillos mientras proteges tus varas.

Reglas: Un botón lanza un disco. Los robots de empuje también lanzan discos. Los discos destruyen todo excepto al jugador. El jugador puede controlar los discos haciéndolos rebotar.

¿Por qué es divertido? Este juego es fácil de entender y es similar al juego clásico Breakout. También es divertido porque es de alta velocidad. Este juego sería mejor si fuese más largo.

Tu Revisión de Juego

Nombre Del Juego: _____

Objetivo: _____

Reglas: _____

¿Por qué es divertido? _____

Módulo 2 Bases de Programación

Actividad 2.1 Introducción al lenguaje de programación Kodu

En esta actividad escribiremos nuestro primer programa de Kodu Game Lab. Kodu Game Lab posee tres tutoriales por defecto y trabajaremos a través de todos ellos.

Demostrar el Tutorial 1 y cómo resolverlo.

Puntos Clave:

Metáforas físicas y declaraciones “When-do” que definen el lenguaje de programación.

Recaltar el menú de ayuda-en-juego el cual muestra cuáles llaves y botones puede ser utilizado para navegación y para acceder a varias pantallas y barras de herramientas.

Notas

Completa el Tutorial 1

1. Abrir el mundo **Tutorial 1**.

El robot Kodu te dice “**Quiero visitar el castillo**”

2. Presiona **Esc** (teclado) o **start** (control) para editar el mundo.

3. Selecciona el ícono “**Agregar y programar objetos**”



de la barra de herramientas.

4. Selecciona el personaje haciendo **click derecho** en él (teclado) o seleccionando Y (panel de juego). Debería estar brillando en amarillo and el menú de edición debe aparecer.

5. Selecciona **Programar** hacienda click derecho en él.


El código del robot debe apreciarse ahora. Tenemos dos opciones, que son editar la línea 1 en el código existente o agregar una nueva línea al código. Editaremos la primera línea pero agregando una nueva línea de código, lo cual es válido también.


La primera línea indica



Debemos cambiarla a



6. Remover los cuadros de expresión y de corazones haciendo click derecho en ellos o presionando  en el panel de juego.

7. Agrega nuevos cuadros con el click izquierdo (o presionando  en el panel de juego) el ícono



a la derecha del cuadro “do”.

8. Selecciona move






9. Selecciona towards.



Ahora el código debe ser:



9. Presiona esc o ( Dos veces) para salir de las pantallas de programación y regresar a la Barra de herramientas.

10. Click en Play  para correr el programa o selecciona  con el panel de control y después

11. El robot Kodu se debe mover hacia el Castillo.

Actividad 2.2 Resolver tutoriales 2 y 3

En esta actividad resolveremos los tutoriales 2 y 3. Las instrucciones para resolver estos retos de programación están incluidos debajo pero primero intente resolverlos sin instrucción primeramente.


Pídale al grupo resolver los tutoriales 2 y 3. Motívelos a resolverlos sin instrucciones primeramente.


Notas


Solución Tutorial 2

Objetivo: Necesitamos reprogramar el robot cycle para que pueda saltar el río.

1. Abrir el Tutorial 2.
2. Presionar **Esc** (teclado) o **start** (control) para editar el mundo.

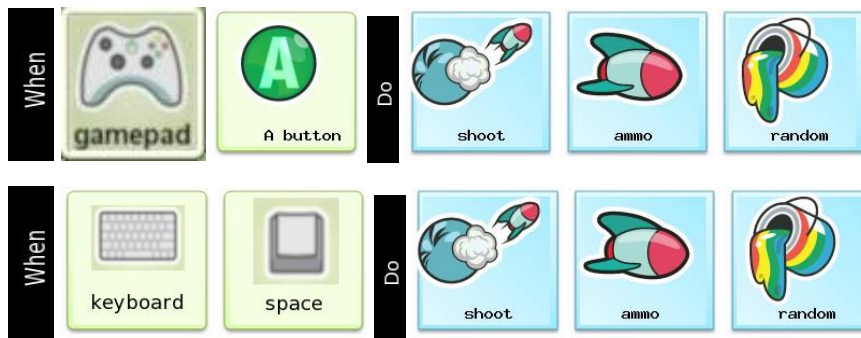
3. Selecciona el ícono “Agregar y Programa Objetos”  de la barra de herramientas.

4. Selecciona el personaje haciendo **click derecho** en él (teclado) o moviendo el cursor  el panel de juego. Debe estar brillando en Amarillo y el menú de edición debe aparecer.

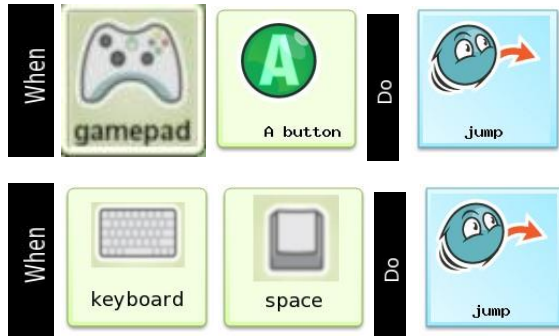
5. Seleccionar **Programar** haciendo click izquierdo o con el botón  del panel de juego.


El código del robot debe mostrarse. Tenemos dos opciones, que son editar la línea 3 y 4 en el código existente o agregar nuevas líneas al código. Editaremos las líneas pero agregando una nueva línea de código, lo cual es válido también.

Las líneas actuales son:





Necesitamos cambiarlas a:



6. Agrega nuevos cuadros haciendo click izquierdo (o presionando  en el panel de control) en el icono



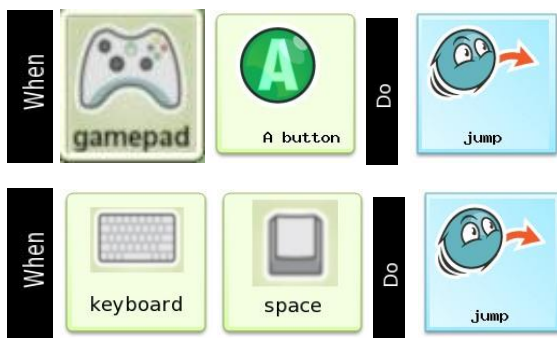
7. a la derecha del cuadro “do”


8. Selecciona  utilizando el botón izquierdo del mouse o presionando  en el panel de juego.




Nota clickea en acciones para encontrar



El código ahora debe ser



8. Presiona esc (o  Dos veces) para salir de la programación y regresar a la Barra de herramientas

9. Click en Play  para correr el programa o selecciona  con el panel de juego y después 

10. El robot Kodu debe saltar con la barra espaciadora o con el botón



Solución de Tutorial 3

Objetivo: Programar el robot cycle para recoger una manzana.


1. Abrir el Tutorial 3.


2. Presionar **Esc** (teclado) o **start** (control) para editar el mundo.

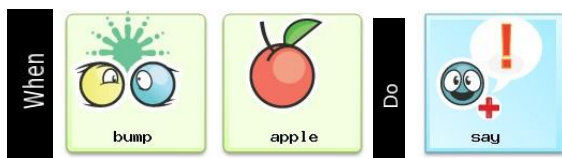
3. Seleccionar el ícono “**Agregar y Programar Objetos**”



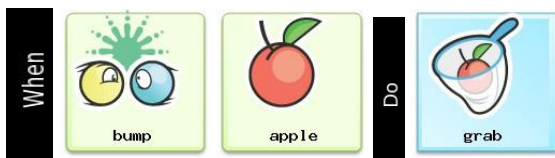
de la barra de herramientas.

4. Seleccione el personaje haciendo **click derecho** en el (teclado) o moviendo el cursor  con el panel de juegos. Debe estar brillando en amarillo y el menú de edición debe aparecer.

5. Seleccionar **Programar** haciendo click izquierdo o con el botón  del panel de juego. La segunda línea de código indica



Necesitamos cambiarla a:



6. Agrega nuevos cuadros con click (o presiona



en el panel de juego) al icono + al lado de “do”.



Pulse

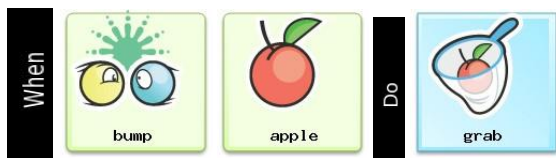


7. Selecciona con el botón izquierdo del mouse o presionando



en el panel de juego. Nótese el mantener oprimido el botón




El código debe ser:



8. Presiona **esc** en el teclado (o



dos veces en el panel) para salir de las pantallas de programación y regresar al menú de edición.

9. Click en Play  Para correr el programa o seleccione  con el panel y después  para jugar el juego

10. El robot Kodu debe ahora recoger la manzana y ser capaz de cruzar el río.

Actividad 2.3 Conociendo los robots y los Objetos

En esta actividad exploraremos varios robots y objetos que están disponibles en Kodu Game Lab. Investigaremos varios atributos y habilidades de los diferentes robots

Utilice robots y Mundo de Objetos para demostrar el rango de los robots y objetos disponibles en Kodu Game Lab.

Puntos principales para recalcar.

1. Diferentes robots tienen diferentes atributos y habilidades
2. Utilice colores para distinguir diferentes roles de personajes, ciclo rojo, ciclo negro.
3. Todos los robots y objetos tienen propiedades que se pueden implementar mientras el juego es creado pero no utilizando programación durante el juego.

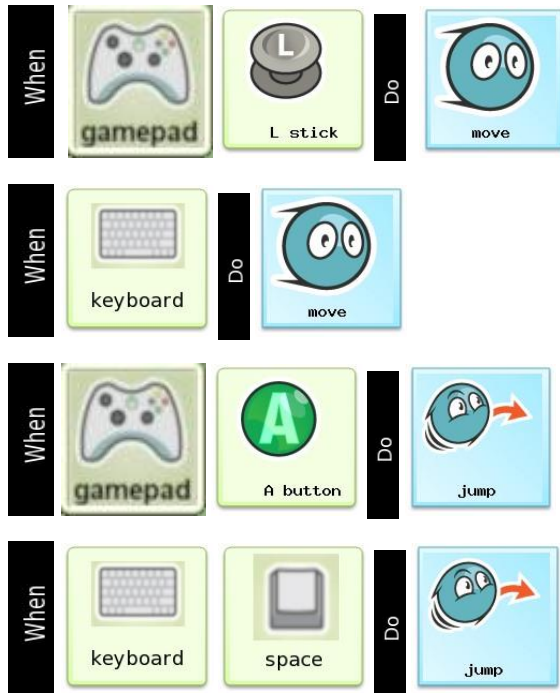


Notas

Actividad 2.4 Programando robots

Robots controlados del Jugador

Hemos visto en los tutoriales previos que podemos escribir programas simples a fin de controlar un robot con el teclado o el control de Xbox. También podemos escribir acciones como saltar con teclas específicas del teclado o el panel de juego.

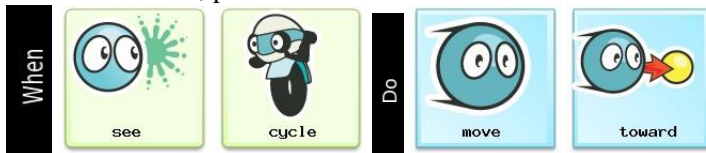


Robots AI

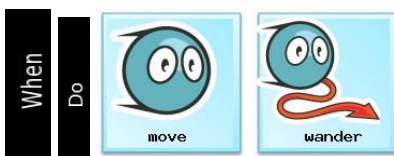
Veamos ahora cómo podemos programar otros robots que interactuarán con el robot del jugador.



Alternativamente, podríamos mirar en vez de escuchar



Si solo queremos que los robots vaguen en vez de seguir, podemos utilizar:



Cuando se tienen sensores físicos:

Puntos Clave:

1. Mirar, escuchar, obtener, dañar, disparar, golpear, sanar, sostener, en tierra y en agua son acciones físicas del robot o el objeto.
2. Temporizador, puntuación, teclado y mouse son acciones externas a las que el robot o el objeto puede responder.
3. Todas las acciones no están disponibles para todos los robots y objetos.



Hacer acciones físicas:



Los robots y los Objetos tienen propiedades que definen ciertos atributos, los cuales determinan cómo aparecen y se comportan. La programación no puede modificar estas propiedades mientras el juego esté corriendo y debe hacerse en el escenario de diseño.

Notas

Actividad 2.4 Crea un Juego de Dos robots

En esta actividad crearemos un simple juego de dos Bot como Tag o similar.

Esta actividad es para crear un simple juego de dos bot. El juego no debe ser complicado pero debe ser alcanzable en un corto periodo de tiempo

Crea tu primer robot.


1. Abrir **Mundo Pequeño con Agua**.


2. Presiona **Esc** (Teclado) o **start** (control) para editar el mundo.

3. Selecciona el ícono “Agregar y Programar Objetos”







de la barra de herramientas.

4. Agrega el personaje haciendo **click izquierdo** en el (teclado) o con el botón  del panel de juego. Selecciona el robot o el objeto que desees.

5. Para programar el objeto selecciona **Programar** con el click izquierdo o con el botón  en el panel de juego.

6. Agrega el código deseado.

7. Presiona **esc** en el teclado (o  dos veces en el panel) para salir de las pantallas de programación y regresar a la barra de herramientas.

8. Click en Play  para correr el programa o selecciona  con el panel de juegos y después  para jugar el juego.

9. Repite el proceso para crear tu segundo robot.

Notas

Módulo 3 Proceso de Diseño Informal

Actividad 3.1 Generando Ideas

Pídale al grupo que piense sobre el juego que crearon o intentaron crear y que piensen sobre cómo lo pueden hacer más interesante.

Pídale que anote esas ideas.

Lista de ideas sobre cómo mejorar el juego.

Actividad 3.2 – Ideas de Juego

En esta actividad generaremos algunas ideas para juegos que podemos hacer con Kodu Game Lab.

Lista una idea de juego que podrías tener:

Ejemplo

Nombre: Duck-Out

Objetivo: Este juego es similar a bonk out excepto que en vez de robots de castillos y barras, se utilizarán botes y tortugas y el juego será jugado en el agua. Las tortugas tendrán la habilidad de sumergirse bajo el agua por dos segundos para evitar los discos.

Reglas: El jugador controla el pez. El pez debe proteger las tres tortugas de los discos voladores rebotándolos. El jugador también puede ocasionar que las tortugas se sumerjan bajo el agua por un corto periodo de tiempo para evitar los discos voladores.

robots utilizados: Botes, tortugas, peces y discos.

Nombre: _____

Objetivo: _____

Reglas: _____

Robots utilizados:

Actividad 3.3 Ejemplos Escolares / Casos de Estudio

En esta actividad daremos un paso atrás de pensar sobre juegos y observar cómo varias escuelas han utilizado Kodu Game Lab en su programa de aprendizaje.

Notas

36 + 36 = 44 + 44 = 20 min
 186 + 186 = 44 + 44 = 20 min
 Time 20 min
 Score Board

KODU GAME DESIGN TEMPLATE

Draw and label your Kodu world.

<p>Describe the Game (and what do the players have to do?)</p> <ol style="list-style-type: none"> 1. Collect Red apples. 2. Answer the questions. 3. Find Bonus stars. 4. Don't get pink apples. 5. Get past bad guys. 6. Don't get shot by cannons. 7. Get star in castle. 8. Tele port to Answer. 	<p>List the Kodu objects and their purpose.</p> <p>Apples: Get points Pink apples: Lose points Stars: Bonus Points Trees: Forest Kodu: controller</p>	<p>Describe how a player wins/finishes the game, how a player scores and how long your game goes for.</p> <p>It goes for 20 mins. you win by collecting stars and apples and you have to go through every stage and to win fully you have to get to the big answer at the end of the game.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Módulo 4 Creando Mundos de Juego

Actividad 4.1 Diseñando Espacios de Juego

Esta actividad crea el entendimiento de consideraciones de recursos, propiedades de mundo y cámaras. Los mundos amplios utilizan una gran cantidad de recursos y es crucial utilizar tan poca tierra como sea posible.

Mostrar el video de Tavish Hill adjuntado en el Power Point del curso



Actividad 4.2 Recetas y Patrones de Diseño

Las recetas y patrones de diseño permiten a los programadores utilizar buenas pruebas y entender acercamientos para resolver problemas. Esta actividad explicará cómo los patrones de diseño pueden ayudar al diseñador del juego a utilizar patrones comprobados para resolver problemas comunes de diseño de juegos. También hará introducción al concepto de instrucciones de código que son soluciones específicas de lenguaje para problemas comunes de programación.

Esta actividad hará introducción de conceptos de patrones de diseño y recetas de código utilizados en el recordatorio de este curso.

Power Point. Motivar el grupo a hacer preguntas si estas surgen.

Puntos clave: Los patrones de diseño son soluciones estándar de documentos para problemas familiares. Las instrucciones son códigos específicos de lenguaje para resolver problemas de programación.

Actividad 4.3 Patrones de Espacio de Juego


Esta actividad hará introducción a algunos Patrones de Diseño de Mundo de Juego. Cada patrón tiene una diapositiva de descripción en el Power Point del Curso y también tiene por lo menos un juego de demostración.

Patrón de Diseño: Mundo Abierto



Los juegos que utilizan un patrón de diseño de mundo abierto permiten que el jugador pueda recorrer donde quiera. Los juegos de mundo abierto usualmente envuelven una Jugabilidad no lineal con jugadores capaces de elegir la secuencia en la que completan las tareas del juego. La cámara sigue al jugador (la cual está por defecto con Kodu Game Lab) y por lo tanto los juegos de mundo abierto son más aprovechados por juegos de un solo jugador.

Puntos Claves: Un Mundo abierto soporta una jugabilidad abierta. Algunos juegos no tienen ninguna tierra. Los mundos abiertos son más ajustables a juegos de estrategia donde la Jugabilidad no es lineal.

Notas

	<p>Utiliza la Herramienta de Borrar Para remover robots y objetos de tu mundo de manera rápida</p>
-----------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------

Cuando se edita un mundo, utilice F3 para encender el **Complemento a la red** para que el tamaño de la brocha se mantenga constante, las coordenadas del cursor se mostraran en la esquina inferior izquierda de la pantalla.

Presione  y  o el scroll del mouse para hacer zoom hacia adentro o hacia afuera. El zoom puede ser útil para encontrar el cursor si este está escondido detrás de la tierra.

Importante: Al momento de crear los mundos, monitoree el medidor de recursos y asegúrese de que no entre en el área roja. Esto es particularmente importante cuando se crean amplios mundos abiertos.

Patrón de Diseño: Carrera

Los juegos de carreras requieren que el jugador corra alrededor de una pista predefinida o espacio. Usualmente hay puntos de marca, con la meta de vencer a los oponentes o registrar el tiempo más rápido. Los juegos de carreras reflejan las carreras de la vida real de forma muy aproximada y por lo tanto, un patrón simple para entender y para jugar.

Demostrar el juego **Pure Plasma Racing**

Puntos Clave: Los caminos pueden ser utilizados para crear carreteras para personajes controlados por inteligencia artificial.

Notas




Receta: Oponentes de la pista de carreras

Para programar a los robots para que corran contra un jugador, simplemente cree un camino y luego programe el robot para que siga ese camino.

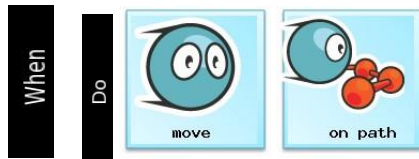
Al utilizar el control de Xbox los caminos son creados con la Herramienta de Objeto.

Al utilizar el teclado y el mouse, el botón de Agregar Camino está en el menú de herramientas.

Selecciona el Grupo de Objeto y luego el tiempo de camino que quieres (plano, carretera, pared o flora).

Utiliza el botón  Para agregar más nodos, mueve el cursor a la siguiente posición deseada y luego presiona  para agregar el nodo. Si la pista de carreras es un bucle que permite múltiples vueltas entonces el último nodo debe ser posicionado encima del primer nodo. Presiona  para terminar.

Coloque el robot en el camino y configure de acuerdo a las siguientes instrucciones.



Diseño de Patrón: Juegos de desplazamiento lateral

Un juego de plataformas es un juego donde el jugador sólo puede moverse en dimensiones 2D, también llamado como un desplazador lateral. Los Juegos de desplazamiento lateral generalmente requieren que el jugador realice un montón de saltos; encima de las lagunas y de otros obstáculos que puedan o no moverse. Estos juegos de desplazamiento lateral eran extremadamente populares en los años 80 y tienen un toque nostálgico cuando se juega.

Puntos clave:

1. Uso de las restricciones Este/Oeste durante el movimiento.
2. Configurar la cámara para fijar el movimiento durante el desplazamiento en la opción configuración del mundo (Muy útil).



Notas

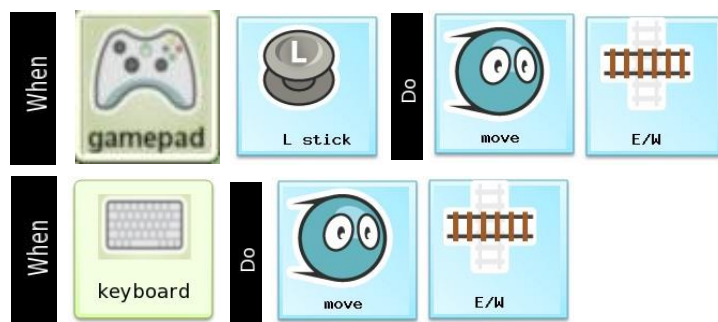
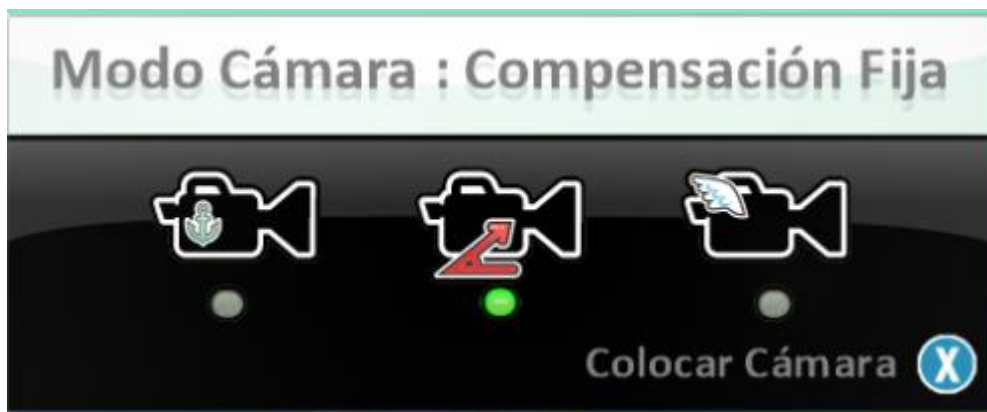
Instrucciones: Desplazamiento Lateral

Por lo general, la creación de un camino en nuestro juego puede ser la mejor opción para crear un juego de desplazamiento lateral, pero también puedes crear un terreno (tierra) en tu plataforma.

Necesitamos la cámara para seguir al personaje del jugador desde un extremo a otro. Para lograr esto vamos a establecer el modo de cámara compensación fija.

Clic en Configuración de mundo:

1. Seleccionar Modo Cámara: Compensación fija.
2. Presiona el botón  para colocar la cámara en su posición de inicio y luego presiona el botón  para establecer la posición.



Diseño del patrón: Laberinto

El uso de un laberinto como su mundo de juego convierte su juego en un rompecabezas que requiere que el jugador use la habilidad de encontrar la ruta correcta. Algunos laberintos tienen una sola ruta que necesita ser descubierto, mientras que otros laberintos tienen múltiples rutas posibles

Puntos clave:

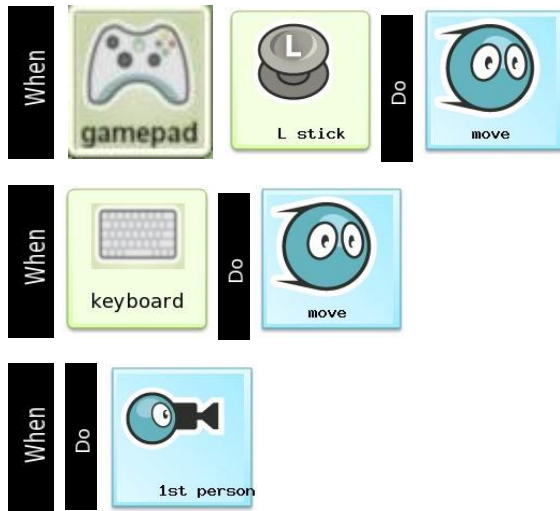
1. Colocar la cámara en posición de la primera persona.

Notas

Instrucciones: Juego del Laberinto

Al hacer un laberinto es más fácil iniciar elevando toda el área de laberinto y luego bajar el laberinto entre las paredes. El aumento de las paredes es mucho más difícil.

Para un juego de laberinto es útil configurar la cámara en compensación fija (para el juego de desplazamiento lateral) detrás robot del jugador o, alternativamente, ajustar la cámara a primera persona. Para configurar la cámara a la primera persona siga las siguientes instrucciones.



Diseño de patrón: Juego de mesa fijo

Algunos espacios de juegos son pequeñas áreas fijas. Estos espacios de juego son más adecuados para los juegos multijugador sin funcionalidad de pantalla dividida. Al tener un espacio y una cámara fija todos los jugadores tienen la misma perspectiva.

Juegos como PacKodu son un buen ejemplo.

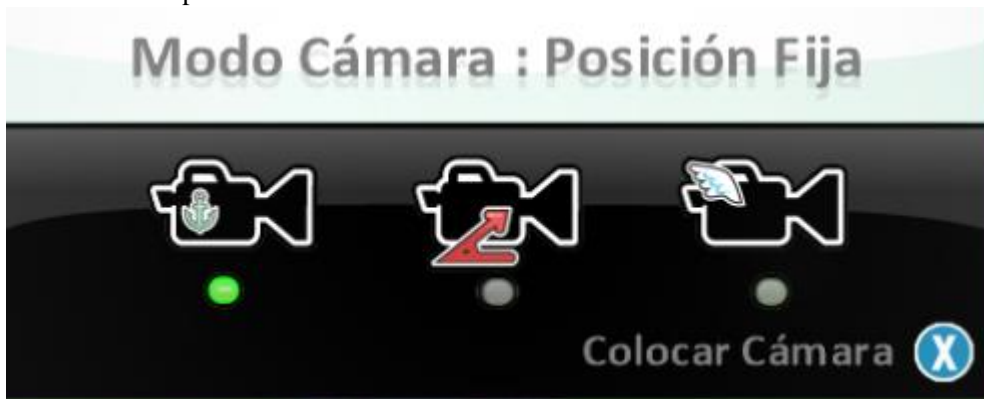
Adecuado para juegos de varios jugadores.

Notas

Instrucciones: Juego de mesa fijo

Necesitamos que la cámara permanezca en la misma posición durante todo el juego. Para ello vamos a establecer el modo de la cámara a la posición fija.

1. Clic en configuración de mundo.
2. Seleccionar Modo de Cámara: Posición Fija
3. Presiona el botón **X** para colocar la cámara en su posición de inicio y luego presiona el botón **A** para establecer la posición.



Module 5 Patrones de diseño del juego

Actividad 5.1 Patrones de progreso del juego

Esta actividad introduce los conceptos avanzados de programación en Kodu, como el uso de las puntuaciones como variables.

Diseño de patrones: Límites de tiempo

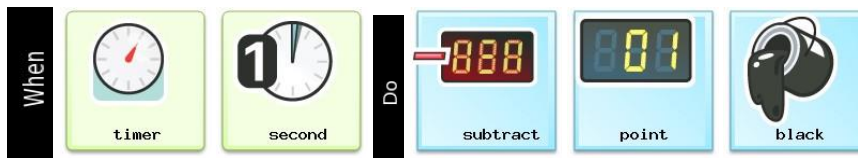
Los límites de tiempo requieren que el jugador complete una acción, alcance una meta o alternatively se establece un límite de tiempo en donde el jugador debe sobrevivir para poder ganar. Los relojes de cuenta regresiva suelen mostrar el tiempo restante para dar al juego una sensación de urgencia. Algunos juegos cuentan con bonificaciones de tiempo que se obtienen a través del alcance de ciertas tareas.

Notas

Instrucciones de Kode: Tiempo límite

Notas:

1. Asegúrese de conectar el temporizador de juego a un robot u objeto que no puede ser destruido durante el juego.
2. Seleccione un color para usar como tu temporizador y no utilice el marcador de color para nada más.
3. Las instrucciones de este ejemplo establecen el tiempo durante 30 segundos y utiliza el color negro para el temporizador. Después de 30 segundos, el jugador gana el juego.
4. Consulte el juego Wack-a-Lama para ver esta instrucción en acción.



Patrones de diseño: Puntuación

Las puntuaciones son una representación numérica de éxito de un jugador. Los puntos pueden ser añadidos a los jugadores cuando logran ciertos objetivos o son deducidos por las actividades en las que han fallado. La puntuación de un jugador por lo general se muestra en todo momento durante un juego, a menudo los puntos conseguidos se muestran como superposición una vez alcanzado el objetivo, con la finalidad de añadir la sensación de logro y progreso.

Notas

Instrucciones de Kode: Puntuaciones

Notas:

1. Esta instrucción añade 1 punto a la puntuación de azul cada vez que un balón golpea el dirigible y 10 puntos cada vez que un platillo es golpeado. Cuando la puntuación es de más de 100 el jugador gana el juego.
2. Programar estas instrucciones al robot que está haciendo los disparos.
3. Esta instrucción funciona tanto con el teclado y el gamepad.
4. Consulte Xevon v06 para ver estas instrucciones.



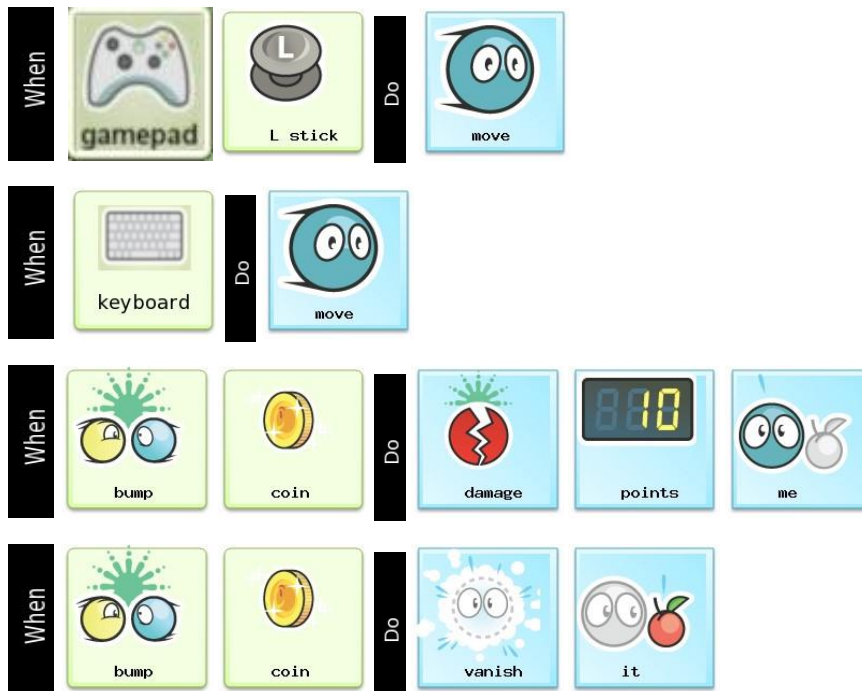
Patrón: Bienestar

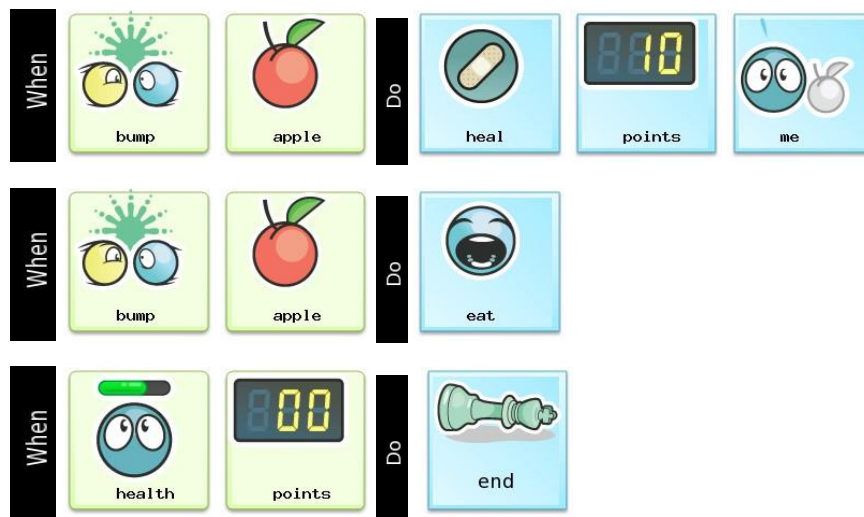
El bienestar del robot jugador también se puede utilizar para indicar el progreso en el juego. La barra de bienestar muestra la condición actual del personaje y proporciona información inmediata al jugador.

Instrucciones: Salud

Notas:

1. Cuando el robot toca una moneda de 10 puntos se reduce del bienestar del robot y la moneda desaparece.
2. Cuando el robot toca y se come una manzana 10 puntos se añaden a la salud del robot.
3. Cuando la salud del robot es cero el juego termina.
4. El número de puntos por golpe (de salud) con los que un robot inicia se establece en su configuración. Los valores oscilan entre 0 y 1000 en incrementos e incrementos de 5 se pueden configurar.





Patrón: Puntos Guardados

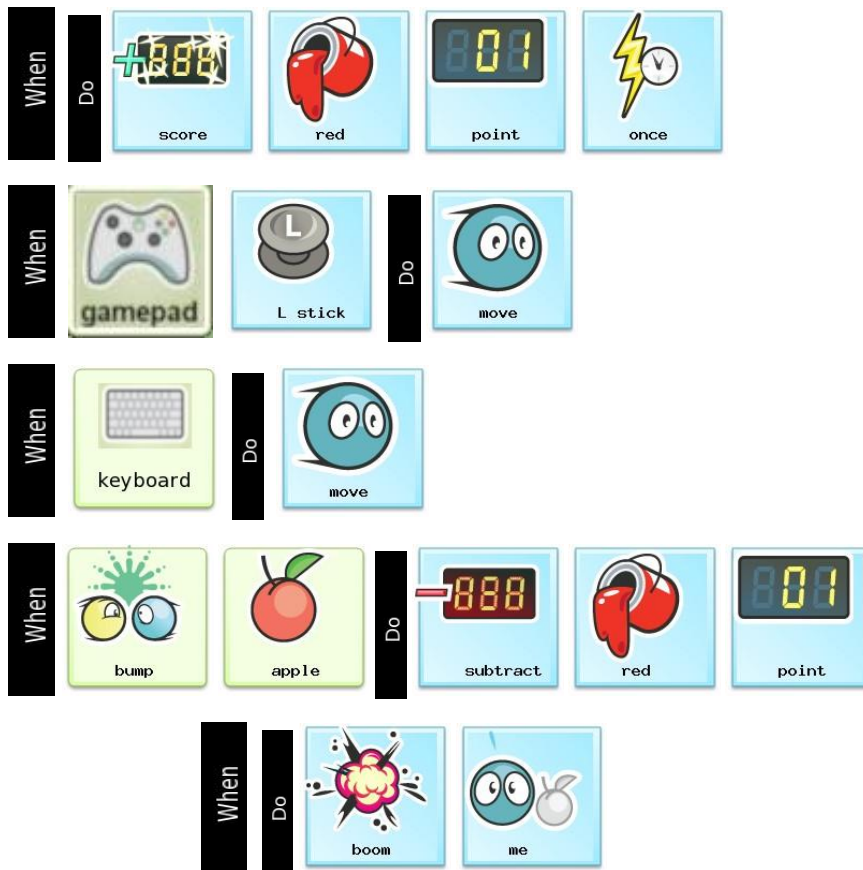
Puntos guardados (también llamados Puntos de comprobación) son puntos de gran ayuda en un juego, por lo general después de haber completado una sección difícil, cuando el juego se guarda automáticamente. Si el usuario falla durante la próxima sección del juego, el juego se reanuda desde el punto de guardado en lugar de que el jugador comience el juego desde el inicio. Esto evita que los jugadores tengan que jugar nuevamente desde el inicio del juego.

Notas

Instrucciones: Puntos Guardados

Notas:

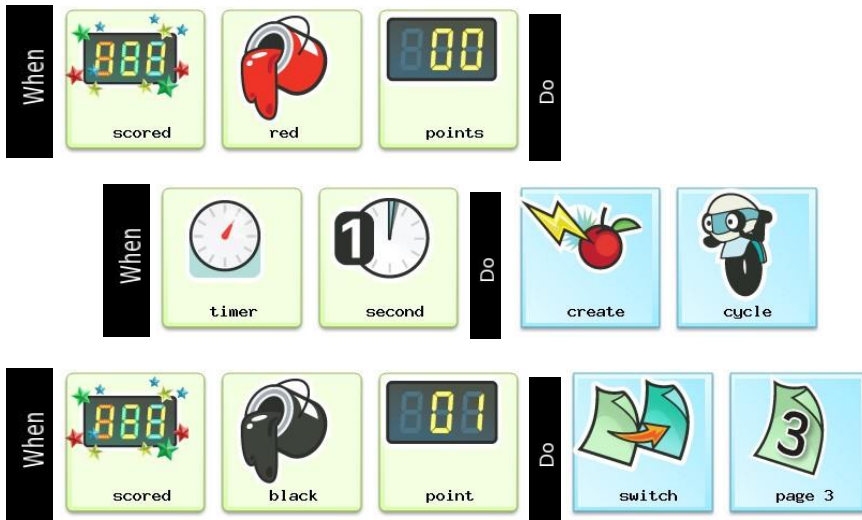
1. El robot Cycle es el personaje del jugador.
2. La puntuación roja se utiliza para dar un seguimiento de la vida de los jugadores. Un valor de 1 indica que el jugador está vivo y jugando mientras que un valor de 0 indica que el robot del jugador debe ser regenerado.
3. La puntuación en negro se utiliza para realizar un seguimiento de los puntos guardados. 0 indica que el jugador está en el comienzo del juego. 1 indica que el jugador ha llegado a guardar 1 punto, y así sucesivamente.
4. Un robot debe ser creado y programado en cada punto guardado, en este ejemplo hemos utilizado robots stick pero el tipo de robot usado no importa.
5. No existe ningún código para la página 3 de los puntos guardados.
6. Utilice la configuración de los puntos guardados para reducir la audición a su mínimo valor, dado que el robot stick solo escuche cuando el robot Cycle este cerca.
7. Una puntuación de color diferente podría ser usado para rastrear y limitar el número de nuevas vidas, esto no se muestra en las instrucciones.



Punto de partida (punto guardado 0)



2

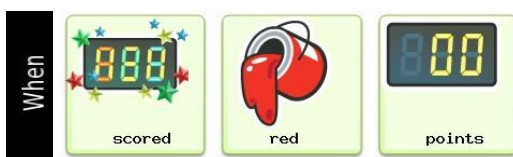


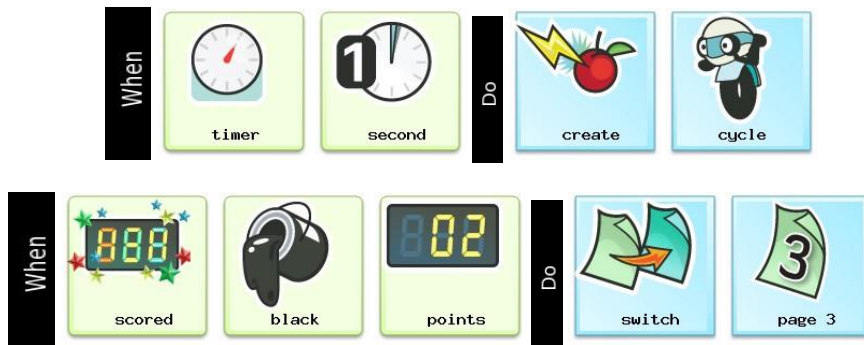
Punto guardado 1

1



2





Los puntos guardados sucesivamente necesitaran que los puntos negros incrementen en 1.

Patrón de diseño: Múltiples niveles

Tener múltiples niveles en un juego es una gran manera de transmitir un sentido de progreso para el jugador. Los juegos también utilizan los niveles para aumentar la dificultad y / o introducir nuevas mecánicas de juego.

Un gran ejemplo del uso de múltiples niveles es halox's Dual y Portal Games, ambos están disponibles como videos de youtube.

Puntos clave: Este es un ejemplo avanzado.

Notas

Instrucciones: Múltiples niveles

Notas:

1. La puntuación de color negro son usados para llevar el seguimiento del nivel en el cual se encuentra actualmente el jugador.
2. El robot del jugador debe ser creado al inicio del juego
3. Los niveles posteriores se pueden hacer copiando el inicio de nivel 2 y el punto final e incrementando los puntajes negros en el código.
4. Establezca el puntaje negro a **Off**





Punto de partida en el nivel 1



Punto de llegada en el nivel 1





Stickbot: Punto de partida en el nivel 2



Stickbot: Punto de llegada en el nivel 2



Actividad 5.2 Patrones de la Jugada

Patrón: Poderes especiales

Los poderes especiales o Power Ups, dan un tiempo de ventaja limitado al jugador que los activa. Estos requieren que los jugadores tomen decisiones estratégicas sobre cuándo utilizarlos, además estos necesitan ser ganados o requieren un tiempo para recargarse.

Los poderes especiales resuelven el problema de tener que completar diferentes tareas o permiten la variación de dificultad al ceder que los usuarios aumenten las habilidades para enfrentar retos más difíciles sin realizar las tareas requeridas.

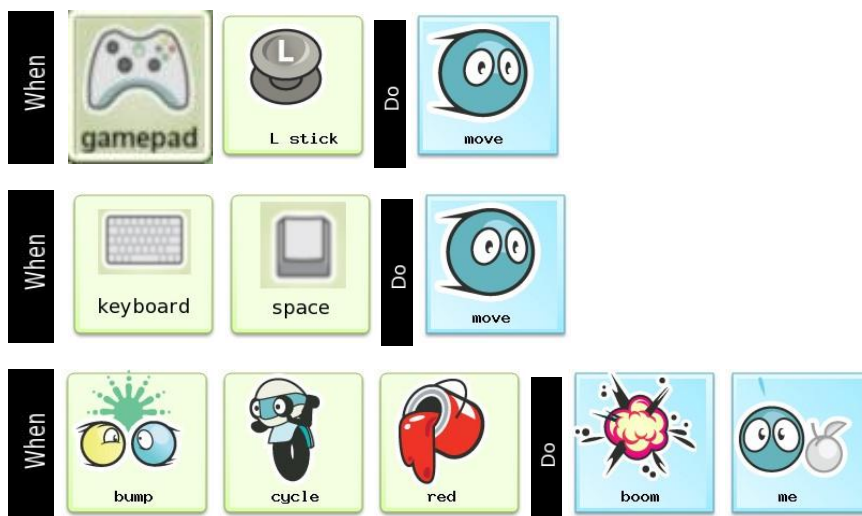
Por ejemplo, un personaje puede volverse invulnerable o ser capaz de moverse más rápido por un período de tiempo después de la obtención de un artículo o después de pulsar una tecla. Los sonidos o efectos visuales se pueden utilizar como señal de que los Power Ups están siendo utilizados y/o a punto de finalizar.

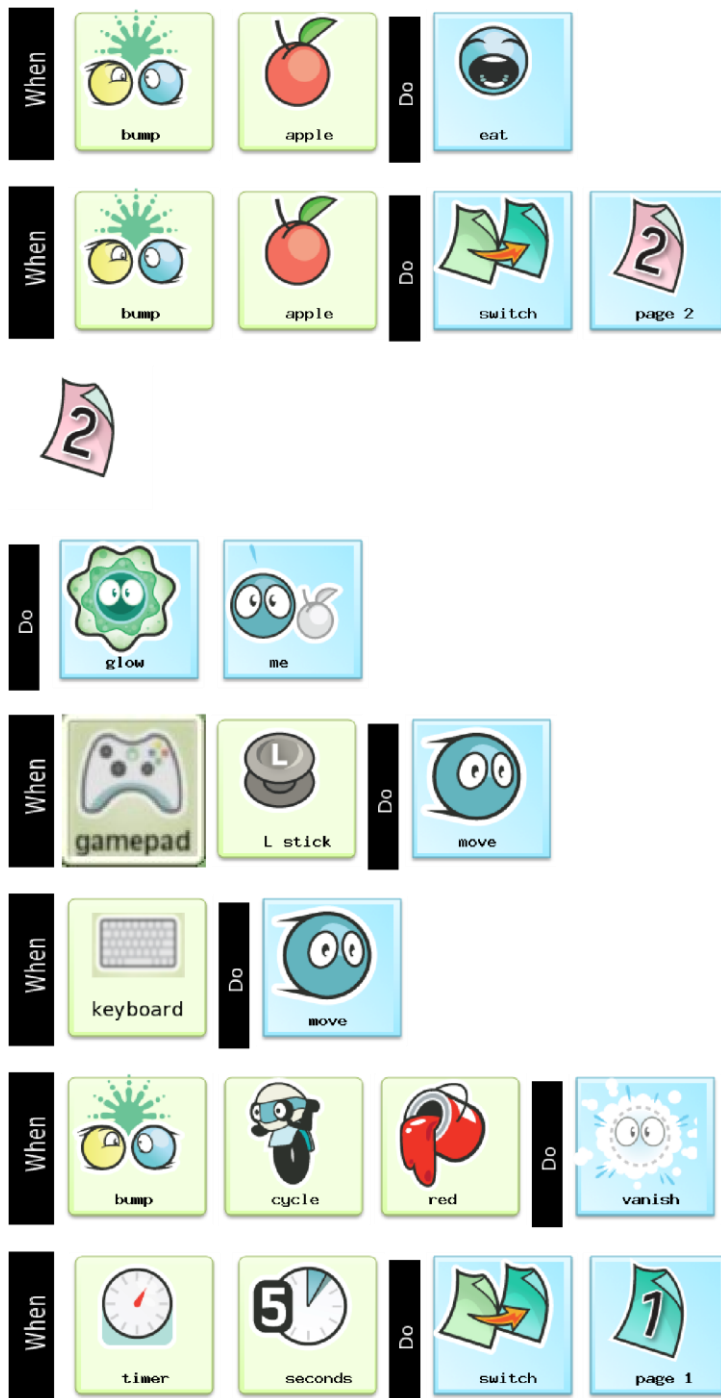
Puntos Claves: Estas instrucciones muestran el uso de páginas y un temporizador.

Notas

Instrucciones: Power Up

Nota: Con estas instrucciones usted va a programar un robot que será controlado por un jugador; el cual perderá el juego cuando tropiece con un robot cycle rojo, cuando el robot choque con una manzana y se la coma este adquirirá un Power Up de manera que los robots cycle serán destruidos cuando sean tocados por el robot del jugador. El Power Up tiene una duración de 5 segundos y luego el robot del jugador vuelve a su estado original.





Patrón: Transferencia de control

Algunos juegos permiten al jugador controlar diferentes personajes en las diferentes etapas del juego. Esto podría incluir cuando el personaje de un jugador entra en un coche o aborda un barco. El nuevo personaje tendría generalmente capacidades diferentes y por lo tanto el juego sería diferente que resulta en un mayor interés para el jugador.

Notas

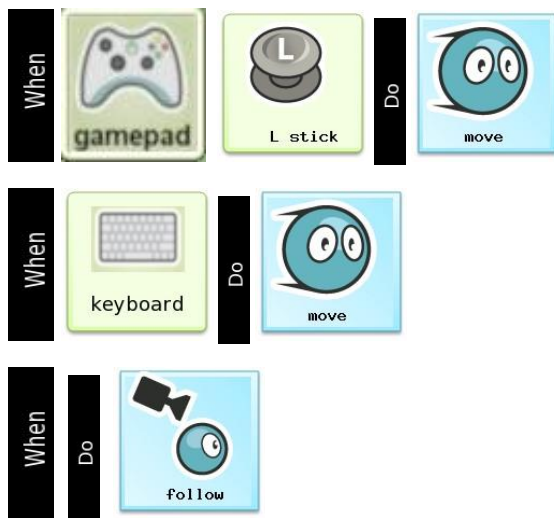
Instrucciones: Transferencia de control

Notas:

1. Este ejemplo muestra como un robot cycle controlado por un jugador puede "entrar en un barco" el cual debe ser controlado por el jugador.



Robot Cycle: Robot inicial controlado por el jugador.

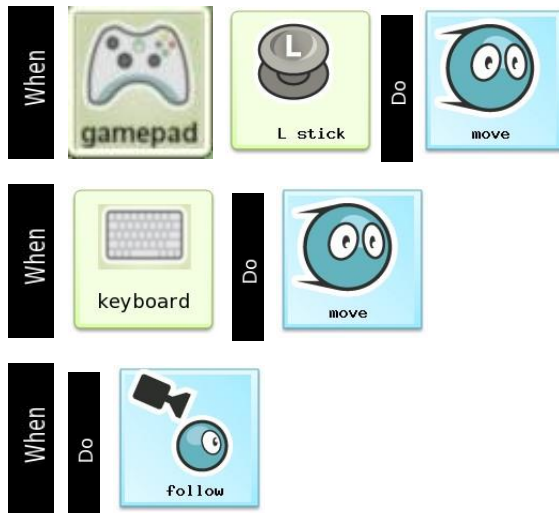


Floatbot: Robot controlado por el jugador una vez abordado por el robot cycle

1



2



Patrón: The Big Boss

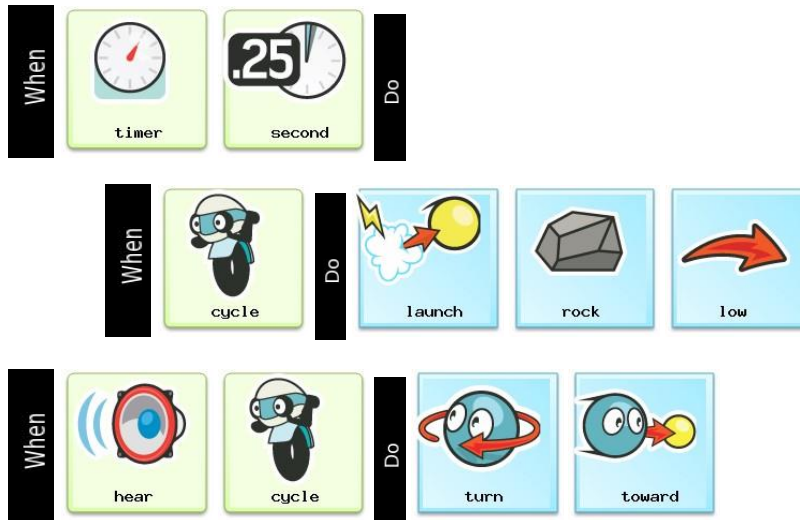
Juegos o niveles en juegos a menudo terminan con una batalla con un gran jefe. El Big Boss es un rival mucho más difícil y con frecuencia tiene habilidades diferentes a los oponentes anteriores. Luchando contra el Big Boss es una gran manera de construir un sentido de progreso y para dar cierre al juego.

Notas

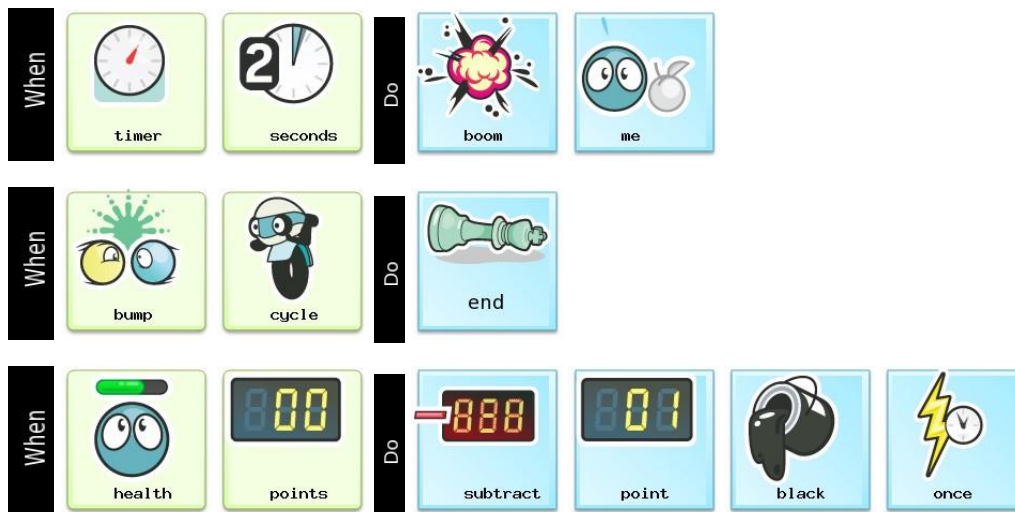
Instrucciones: Big Boss

Notas:

1. Los ajustes del puntaje del Big Boss pueden ser incrementados.
2. Para hacer los ataques más perjudiciales se pueden incrementar los puntos de daño, los cuales son sustraídos de la puntuación.
3. Este solo es un ejemplo de como un Big Boss puede ser creado.



Las rocas son lanzadas por el Boss.



Actividad 5.3 Patrón de diseño de información del juego

Es importante comunicarse con los jugadores para que entiendan claramente el estado del juego y su progreso dentro del mismo. La información del juego se puede comunicar con texto, sonidos, puntuación y cambios en el espacio de juego. La comunicación clara ayuda a evitar la frustración y los resultados inesperados para el jugador.

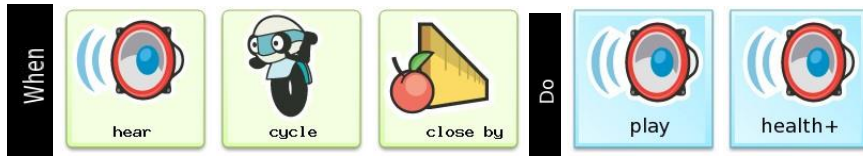
Patrón: Alarma

Las alarmas pueden ser utilizadas en los juegos para advertir a los jugadores del peligro. Estas pueden ser utilizadas para indicar que una fase del juego está a punto de comenzar o terminar. Diferentes sonidos de alarmas pueden utilizarse para notificar al jugador de los diferentes eventos.

Notas

Instrucciones: Alarmas

Estas instrucciones muestran como una alarma puede ser utilizada para indicar que un robot Cycle está cerca de algún objeto.



Patrón: Información del juego

Los cuadros de diálogo se puede utilizar para mostrar información al usuario, dicha información puede simplemente notificar al usuario de su progreso en el juego o puede proporcionar información de que el jugador necesita completar con éxito el juego.

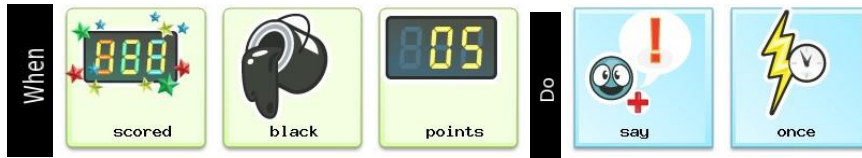
Notas

Instrucciones: Diálogo

Hay tres tipos de cuadros de diálogo de Kodu Game Lab: pantalla completa y dos Globos, una en la que se visualizan secuencialmente las líneas y una en la que se muestran líneas al azar.

Pantalla completa se utiliza para dar mensajes generales al jugador, mientras que los cuadros de diálogo de voz se utilizan cuando caracteres específicos son utilizados para proporcionar información al jugador.

En este ejemplo se mostrará un cuadro de diálogo de pantalla completa cuando el marcador negro llega a cinco puntos.



Use el botón  Luego selecciona el ícono  Para seleccionar el tipo de cuadro de dialogo a utilizar

Module 6 Enfoque en el diseño del juego

Actividad 6.1

En este ejemplo vamos a investigar los procesos de diseño adecuado para el uso de sus estudiantes.

Jesse Schell dice, "piensa en un problema, identifica la mejor manera de resolver el problema, implementa, prueba."

Algunos consejos:

Planifique su juego en el papel.

Comience con un juego simple y luego expándalo.

Probar, probar y probar.

Describe el proceso de diseño que va a utilizar con sus alumnos

Actividad 6.2

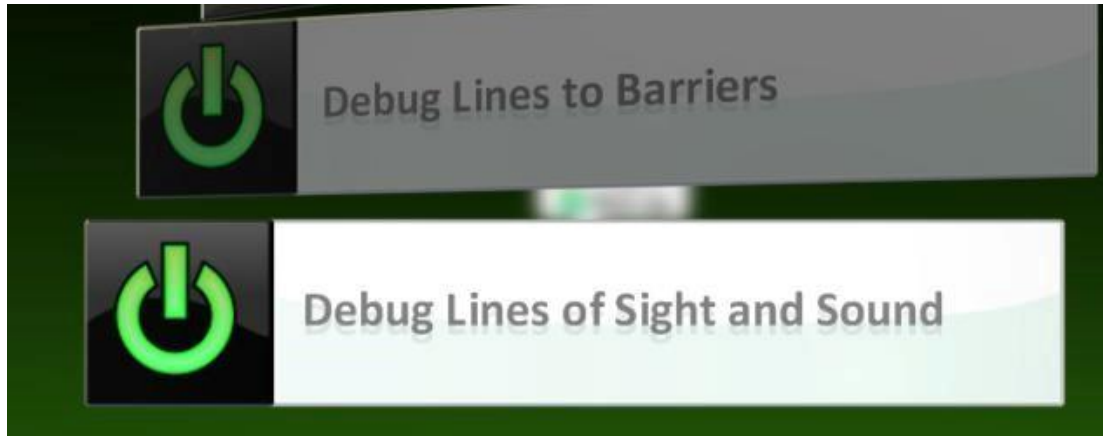
En esta actividad vamos a investigar algunas de las funcionalidades de Kodu Game Lab que hace la vida más fácil para los diseñadores de juegos.

Identificando riesgos: ¿Podemos hacer esto en Kodu?

Hay muchas cosas que son posibles o imposibles de crear con Kodu Game Lab. En el diseño de un nuevo juego, es crucial que el diseñador identifique las partes del juego que pueden no ser posible y los crea primero, de esa manera si el juego es imposible de crear, el tiempo se ha desperdiciado.



Depuración: ¿Por qué no funciona?

Cuando el juego no está funcionando adecuadamente hay líneas de depuración que se pueden habilitar. Como su nombre indica las líneas de depuración de imágenes y sonidos mostrarán a donde el robot puede y no puede ver ni escuchar y es muy útil en el descubrimiento de cualquier código que no esté funcionando correctamente.




Copiar, Pegar, Clonar: ¿Hay una manera más rápida de hacer esto?

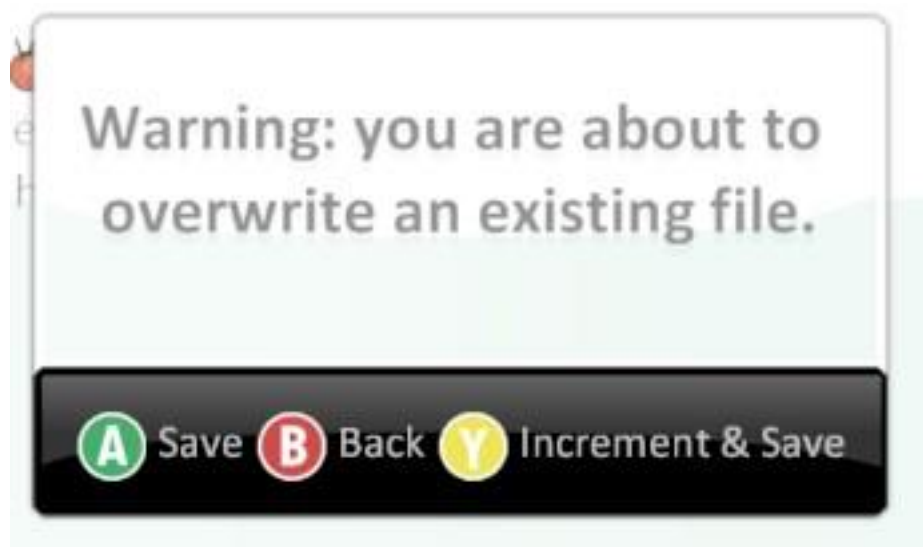
Al crear múltiples robots con la misma programación es a menudo más rápido y más fácil de clonar y pegar el robot u objeto. Los robots y los objetos también se pueden cortar y pegar entre proyectos.

Usa la tecla  para cortar y  para clonar y pegar o alternativelymente con el mouse haz clic derecho en el objeto.

El uso de versiones: ¿Vaya mi juego ya no funciona!?

Al guardar cambios importantes en los juegos con Kodu Game Lab es muy útil para guardarlos como una nueva versión. Esto crea copias de seguridad de su juego lo cual es útil en el momento que hagas cambios y esto provoque que tu juego deje de funcionar. Al guardar una nueva versión de un juego, Kodu le advertirá y

te sugiere que lo guardes como un incremento del juego anterior, presiona  para hacerlo.



Utilizando el Medidor de recursos: ¿Por Qué mi juego está corriendo tan lento?

Kodu Game Lab es para hacer pequeños juegos y Cuando los juegos se tornan demasiado grandes Kodu se hace más lento, a menudo haciendo que el juego no sea agradable e imposible de jugar.

El medidor de recursos se puede visualizar al habilitarlo en la sección Configuración de mundo.



Asegúrese de que el medidor de recursos no llegue a alcanzar el color rojo mientras esté probando tu juego. Algunas veces su código puede ser modificado para limitar los recursos utilizados, por ejemplo, reduciendo al mínimo los comandos para escuchar, pero por lo general será necesario reducir la cantidad de tierra utilizada en el juego.



Modulo 7 Próximos pasos

Discuta con el grupo cómo las escuelas han utilizado Kodu Game Lab y cómo van a introducirlo en sus escuelas

¿Cómo planea usar Kodu Game Lab en su escuela?

Recursos Kodu

Páginas web

<http://fuse.microsoft.com/kodu>

<http://planetkodu.com>

<http://boards.kodux.com>

Todos los materiales del taller se pueden descargar desde

<http://media.planetkodu.com/workshop/resources.html>